LEARN C++
Skill up with our free tutorials

0.3 — Introduction to C/C++

ALEX[1] AUGUST 17, 2023

Before C++, there was C

The C language was developed in 1972 by Dennis Ritchie at Bell Telephone laboratories, primarily as a systems programming language (a language to write operating systems with). Ritchie's primary goals were to produce a minimalistic language that was easy to compile, allowed efficient access to memory, produced efficient code, and was self-contained (not reliant on other programs). For a high-level language, it was designed to give the programmer a lot of control, while still encouraging platform (hardware and operating system) independence (that is, the code didn't have to be rewritten for each platform).

C ended up being so efficient and flexible that in 1973, Ritchie and Ken Thompson rewrote most of the Unix operating system using C. Many previous operating systems had been written in assembly. Unlike assembly, which produces programs that can only run on specific CPUs, C has excellent portability, allowing Unix to be easily recompiled on many different types of computers and speeding its adoption. C and Unix had their fortunes tied together, and C's popularity was in part tied to the success of Unix as an operating system.

In 1978, Brian Kernighan and Dennis Ritchie published a book called "The C Programming Language". This book, which was commonly known as K&R (after the authors' last names), provided an informal specification for the language and became a de facto standard. When maximum portability was needed, programmers would stick to the recommendations in K&R, because most compilers at the time were implemented to K&R standards.

In 1983, the American National Standards Institute (ANSI) formed a committee to establish a formal standard for C. In 1989 (committees take forever to do anything), they finished, and released the C89 standard, more commonly known as ANSI C. In 1990 the International Organization for Standardization (ISO) adopted ANSI C (with a few minor modifications). This version of C became known as C90. Compilers eventually became ANSI C/C90 compliant, and programs desiring maximum portability were coded to this standard.

In 1999, the ISO committee released a new version of C called C99. C99 adopted many features which had already made their way into compilers as extensions, or had been implemented in C++.

C++

C++ (pronounced “see plus plus”) was developed by Bjarne Stroustrup at Bell Labs as an extension to C, starting in 1979. C++ adds many new features to the C language, and is perhaps best thought of as a superset of C, though this is not strictly true (as C99 introduced a few features that do not exist in C++). C++’s claim to fame results primarily from the fact that it is an object-oriented language. As for what an “object” is and how it differs from traditional programming methods, well, we’ll cover that in later chapters.

C++ was standardized in 1998 by the ISO committee (this means the ISO standards committee approved a document describing the C++ language, to help ensure all compilers adhere to the same set of standards). A minor update to the language was released in 2003 (called C++03).

Five major updates to the C++ language (C++11, C++14, C++17, C++20, and C++23) have been made since then, each adding additional functionality. C++11 in particular added a huge number of new capabilities, and is widely considered to be the new baseline version of the language. Future upgrades to the language are expected every three or so years.

Each new formal release of the language is called a **language standard** (or **language specification**). Standards are named after the year they are released in. For example, there is no C++15, because there was no new standard in 2015.

C and C++’s philosophy

The underlying design philosophy of C and C++ can be summed up as “trust the programmer” -- which is both wonderful and dangerous. C++ is designed to allow the programmer a high degree of freedom to do what they want. However, this also means the language often won’t stop you from doing things that don’t make sense, because it will assume you’re doing so for some reason it doesn’t understand. There are quite a few pitfalls that new programmers are likely to fall into if caught unaware. This is one of the primary reasons why knowing what you shouldn’t do in C/C++ is almost as important as knowing what you should do.

Q: What is C++ good at?

C++ excels in situations where high performance and precise control over memory and other resources is needed. Here are a few common types of applications that most likely would be written in C++:

- Video games
- Real-time systems (e.g. for transportation, manufacturing, etc…)
- High-performance financial applications (e.g. high frequency trading)
- Graphical applications and simulations
- Productivity / office applications
- Embedded software
- Audio and video processing
- Artificial intelligence and neural networks

Q: Do I need to know C before I do these tutorials?

Nope! It's perfectly fine to start with C++, and we'll teach you everything you need to know (including pitfalls to avoid) along the way.

Once you know C++, it should be pretty easy to learn standard C if you ever have the need. These days, C is mostly used for niche use cases: code that runs on embedded devices, when you need to interact with other languages that can only interface with C, etc… For most other cases, C++ is recommended.

Next lesson

0.4 Introduction to C++ development

3

Back to table of contents

4

Previous lesson

0.2 Introduction to programming languages

5

6

B U URL INLINE CODE C++ CODE BLOCK HELP!

Leave a comment...

Name*

Email*

Notify me about replies: 🔔

POST COMMENT

Find a mistake? Leave a comment above!

Avatars from https://gravatar.com/[7] are connected to your provided email address.

446 COMMENTS Newest ▾

IceFrog
April 4, 2024 2:50 pm

Hi Alex, what are your opinions on Rust?

0 Reply

> **Alex** Author
> Reply to IceFrog 8 April 5, 2024 9:09 am
>
> Rust is cool, and incorporates a lot of learnings that have happened in language design over the years. I'm not sure whether the benefits are worth the costs though. C++ also has a larger community and decades of library development to draw from.
>
> 0 Reply

Muhammad Essameldin
March 26, 2024 4:33 am

pretty good introduction, well done

3 Reply

Rizi
March 18, 2024 1:58 am

I enjoyed reading this part; it is a concise and engaging chronology and overview.

1 Reply

pat
March 5, 2024 9:50 am

very interesting

0 Reply

by88
February 26, 2024 11:10 pm

appreciate the great content

1 Reply

Ravi
February 24, 2024 5:41 am

Hello from Brazil! Let's learn C++!!

👍 2 Reply

Aron
Reply to Ravi 9 March 16, 2024 1:54 pm

BRAZIL MENTIONED!!!!

👍 2 Reply

RamonVN
Reply to Ravi 9 March 1, 2024 3:00 pm

Esse site é ouro para quem quer aprender a mamar o C++.

👍 1 Reply

Ravi
Reply to RamonVN 10 March 1, 2024 3:14 pm

Sim! C++ do jeito "certo" e aprofudando. Estou amando

👍 0 Reply

abc
February 20, 2024 12:43 pm

Hii

👍 0 Reply

Dean
February 6, 2024 12:03 am

Who else is reading this as the C++ compiler downloads??

👍 1 Reply

Gianni
February 5, 2024 11:14 am

Lessgo

👍 1 Reply

shygosh
February 4, 2024 1:21 am

You know something is of a good quality when there is background story. It helps you fall in love with what you're learning.

👍 7 Reply

方，源
January 28, 2024 12:06 am

3

👍 1 Reply

Kamil G.
January 24, 2024 5:00 am

Okay

👍 1 Reply

Joaquin
January 23, 2024 3:26 pm

based and c pilled

👍 2 Reply

S.E.
January 23, 2024 5:37 am

Boy, what are the C++ people going to do when it gets an update in the year 2103?

👍 2 Reply

> **bewf**
> Reply to S.E. 11 April 9, 2024 8:15 am
>
> think about the year 3103

0 Reply

Alex Author

Reply to S.E.[11] January 23, 2024 10:35 am

C++103 probably.

7 Reply

Wafa Ayan

January 18, 2024 9:28 pm

You are great Alex

1 Reply

IceFloe

January 18, 2024 1:16 pm

Please tell us what a unified set of standards for compilers is, what is included there? And is it possible to find this for review?

1 Reply

Alex Author

Reply to IceFloe[12] January 19, 2024 9:07 am

There is no "unified" set of standards. We talk more about language standards in lesson https://www.learncpp.com/cpp-tutorial/configuring-your-compiler-choosing-a-language-standard/, and there is a link there to one of the C++ standards documents so you can see what it looks like.

4 Reply

IceFloe

Reply to Alex[13] January 20, 2024 4:25 pm

Thanks)))

0 Reply

spanning_tree

January 17, 2024 4:26 pm

Alright

👍 0 Reply

joe
January 7, 2024 8:15 pm

Nice

👍 3 Reply

> **joe #1 fan**
> Reply to joe [14] January 17, 2024 1:15 pm
>
> actual king
>
> 👍 1 Reply
>
> > **joe actual #1 fan**
> > Reply to joe #1 fan [15] January 18, 2024 4:15 am
> >
> > hey who the ^%$# are you buddy >:/
> >
> > *Last edited 2 months ago by Alex*
> >
> > 👍 1 Reply
> >
> > > **Joe #1 fan frfr**
> > > Reply to joe actual #1 fan [16] January 19, 2024 5:56 pm
> > >
> > > stay away from my man cuh
> > > hes mine
> > >
> > > 👍 1 Reply
> > >
> > > > **joe actual #1 fan**
> > > > Reply to Joe #1 fan frfr [17] January 20, 2024 6:28 pm
> > > >
> > > > fake and wrong, I'M the one who feeds him grapes
> > > >
> > > > 👍 0 Reply
> > > >
> > > > > **Joes official actual #1 fan fr fr**
> > > > > Reply to joe actual #1 fan [18] January 29, 2024 5:46 am
> > > > >
> > > > > IM the one who raised him all his life

0 Reply

woh

Reply to Joes official actual #1 fan fr fr 19

March 12, 2024 1:48 pm

wtf is happening here

0 Reply

Justin the Pythoner

January 4, 2024 4:45 pm

Excited to refresh myself on C++ and the new features!

0 Reply

Peter Han

December 31, 2023 7:18 am

This is quite exciting, best of luck to everyone here!

5 Reply

Hadi

December 29, 2023 3:29 am

This is just great! Whoever explained this knows C++ more than the creators lol

0 Reply

Ryan (python coder)

🕒 December 24, 2023 12:11 pm

Fun fact: I'm learning this for game hacking.

I've learned how to use cheat engine already, but I want to step up my game. I don't usually hack in games, but it's always took an interest in me.

I've hacked games like PC Building simulator and cookie clicker. I hope to expand my knowledge to more well played, commonly known games such as "Fortnite", "Call of Duty" etc.

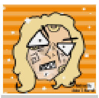
👍 0 Reply

Anant

🕒 December 23, 2023 11:49 am

LETSSS GET ITTT

👍 4 Reply



Aiden T

🕒 December 18, 2023 12:04 pm

WOOHOO STaying focused I recommend hyper bold for any adhd readers

👍 0 Reply

AbeerOrTwo

🕒 December 13, 2023 6:33 pm

Gonna leave a comment after every lesson for motivation

👍 7 Reply

amir

🕒 December 7, 2023 12:42 am

good, this tutorial is best.

👍 0 Reply

Hannah Norman

🕒 December 2, 2023 11:14 pm

another step forward to creative chaos

👍 4 Reply

Harry Larubi

Harry Larubi
December 1, 2023 8:01 pm

lets go

👍 2 Reply

Daniel
November 25, 2023 7:58 am

LETS KEEP AT IT 25/11/2023

👍 1 Reply

Peter
November 22, 2023 6:14 pm

Another step forward, friends. Lets keep at it!

👍 1 Reply

Links

1. https://www.learncpp.com/author/Alex/
2. https://ikeamenu.com/humix/video/Ipms8sWsckf
3. https://www.learncpp.com/cpp-tutorial/introduction-to-cpp-development/
4. https://www.learncpp.com/
5. https://www.learncpp.com/cpp-tutorial/introduction-to-programming-languages/
6. https://www.learncpp.com/introduction-to-cplusplus/
7. https://gravatar.com/
8. https://www.learncpp.com/cpp-tutorial/introduction-to-cplusplus/#comment-595447
9. https://www.learncpp.com/cpp-tutorial/introduction-to-cplusplus/#comment-593996
10. https://www.learncpp.com/cpp-tutorial/introduction-to-cplusplus/#comment-594189
11. https://www.learncpp.com/cpp-tutorial/introduction-to-cplusplus/#comment-592709
12. https://www.learncpp.com/cpp-tutorial/introduction-to-cplusplus/#comment-592536
13. https://www.learncpp.com/cpp-tutorial/introduction-to-cplusplus/#comment-592577

14. https://www.learncpp.com/cpp-tutorial/introduction-to-cplusplus/#comment-591927
15. https://www.learncpp.com/cpp-tutorial/introduction-to-cplusplus/#comment-592471
16. https://www.learncpp.com/cpp-tutorial/introduction-to-cplusplus/#comment-592517
17. https://www.learncpp.com/cpp-tutorial/introduction-to-cplusplus/#comment-592602
18. https://www.learncpp.com/cpp-tutorial/introduction-to-cplusplus/#comment-592640
19. https://www.learncpp.com/cpp-tutorial/introduction-to-cplusplus/#comment-593015